**Questions**

1. Cosa si intende per database?

Un database è un mero contenitore di informazioni e dati strutturati. In particolare, una sua caratteristica è quella di minimizzare lo spazio.

1. Cos’è un DBMS?

Un DBMS (database Management System) serve a controllare un database ovvero è un software di database che serve alla gestione del database. In particolare, consente di creare la struttura del database e permettere agli utenti finali di visualizzare solo la rappresentazione logica, mettendo a disposizione su richiesta le informazioni presenti nel database. Inoltre, deve garantire la condivisione dei dati, dei database persistenti, la privacy ed essere affidabile, efficiente ed efficace.

1. Indica le principali clausole di uno statement SELECT in ordine di esecuzione logica. Descrivi per ciascuna delle clausole indicate la logica di funzionamento.

In ordine di esecuzione logica, Le clausole di uno statement SELECT sono:

* *from* attraverso cui si indica la sorgente dati da interrogare;
* *join* può essere left, right, inner, full outer e attraverso essa si recuperano dati da un’ulteriore sorgente in base alle relazioni con la precedente;
* *Where* condizione di ricerca ovvero filtro da applicare alla tabella restituita dalla precedente clausola (from). Tale clausola utilizza gli operatori di confronto base (= uguale, > maggiore,< minore, <= minore uguale, >= maggiore uguale, != o <> diverso) e gli operatori logici di like (alternativo all’uguale, ma diversamente da quest’ultimo permette l’utilizzo di caratteri jolly quali % che rappresenta 0 o più caratteri e \_ che rappresenta un carattere), between (restituisce i valori che corrispondo al range verificato), or (restituisce i valori che corrispondono a una delle due condizioni), and (restituisce i valori che corrispondono ad entrambe le condizioni), is null/not null (permette di filtrare i valori non conosciuti, unknow o null) e in (restituisce i valori corrispondenti all’elenco). Esso non può essere usato con le funzioni di aggregazione in quanto viene eseguito prima del group by;
* *Group by* consente di raggruppare le righe della tabella filtrata per ogni combinazione univoca indicata nella group by list;
* *Having* è la condizione o filtro da applicare al raggruppamento;
* *Order by* consente di ordinare i dati della tabella restituita in ordine decrescente o crescente;
* *Select* consente di selezionare i campi che saranno restituiti dalla query.

1. Descrivi, immaginando uno scenario a te familiare, il concetto di group by. Utilizza l’approccio che ritieni più efficiente per trasmettere il concetto (suggerimento: disegna anche una sola tabella in Excel o in word con poche colonne e pochi record e descrivi, basandosi sulla tabella stessa, un esempio di group by).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IDLibro** | **Titolo** | **Autore** | **Genere** | **AnnoPubblicazione** |
| 1 | Tra le stupide righe | Alex Coman | Narrativa | 2020 |
| 2 | La magia del ritorno | Nicholas Sparks | Romance | 2020 |
| 3 | Bianca come il latte rossa come il sangue | Alessandro D’Avenia | Narrativa | 2010 |
| 4 | Scheletri | Zerocalcare | Fumetto | 2020 |
| 5 | L’amica geniale | Elena Ferrante | Narrativa | 2011 |
| 6 | Zoo | Paola Barbato | Thriller | 2019 |

Si vuole raggruppare i libri per genere. In particolare, si vuole sapere per ogni genere quanti libri sono presenti nella libreria. La tabella visualizzata sopra ha il nome di Libri.

Al fine di ottenere il risultato richiesto occorre indicare la sorgente dati che si vuole indicare quindi *from Libri,* successivamente raggruppare i libri in base al genere pertanto *group by* Genere, infine con la funzione select si esegue il conteggio dei libri e si indicano i campi che si vogliono visualizzare ovvero *select Genere, Count (IDLibro) as* Conteggio (si utilizza as in modo da rendere chiaro il comando eseguito con la funzione Count).

La query pertanto è questa:

*Select Genere, Count(IDLibro) as conteggio*

*From Libri*

*Group by Genere*

1. Descrivi la differenza tra uno schema OLTP e uno schema OLAP.

L’OLTP (Online Transaction Processing) serve per la gestione del dato e delle transizioni. In particolare, è una base date strutturata in modo da ottimizzare le operazioni che gli utenti operano sul database tramite un software. Tali operazioni, dette CRUD (Create, read, update, delete) sono ottimizzate attraverso la riduzione dello spazio occupato e garantendo l’integrità referenziale. Ciò è garantito attraverso la normalizzazione: suddividere le informazioni in diverse tabelle in modo da non avere ridondanza.

Invece, L’OLAP (online analytical Processing) serve ad ottimizzare l’analisi del dato. In particolare, integra dati provenienti da diverse fonti. A tal fine i dati contenuti sono strutturati in tabelle, ma in tabelle denormalizzate ovvero contenenti più dati e informazioni collegate in un’unica tabella in modo da velocizzare le interrogazioni dei dati.

1. Dato un medesimo scenario di analisi, qual è la differenza in termini di risultato ottenibile tra una join e una subquery?

Se con la subquery si ottengono valori che hanno una corrispondenza tra le due tabelle interrogate allora la join è un sostituto e il risultato ottenuto è il medesimo, in caso contrario no e il risultato non è il medesimo. Pertanto, la differenza.

1. Cosa si intende per DML e DDL?

DML (Data manipulation Language) è un linguaggio che permette di inserire, cancellare e modificare i dati. Invece, il DDL (Data Definition Language) è un linguaggio che permette di agire direttamente sullo schema del database ovvero permette di creare, modificare, eliminare gli oggetti del database.

1. Quali istruzioni possono utilizzare per estrarre l’anno da un campo data? Proponi degli esempi.

Per estrarre l’anno da un campo data si usa Year ().

Esempi:

* Select Year(2022-10.-10 08:10:57.200) restituisce l’anno da una data in formato datatime

Risultato 2022

* Select Year(now()) restituisce l’anno corrente

Risultato 2023

* Select Year(curdate()) restituisce l’anno corrente

Risultato 2023

Si ottiene lo stesso risultato ponendo al posto di curdate() le seguenti funzioni: CURRENT\_DATE, CURRENT\_DATE(), CURRENT\_DATE(n)

* Select Year(2012-10-11) restituisce l’anno da una data specifica in formato date

Risultato 2022

* Select Year(OrderDate) restituisce l’anno di ogni riga della colonna OrderDate

1. Qual è la differenza tra gli operatori logici AND e OR?

or restituisce i valori che corrispondono a una delle due condizioni indicate invece, and restituisce i valori che corrispondono ad entrambe le condizioni.

1. È possibile innestare una query nella clausola SELECT?

Sì, se voglio visualizzare anche un campo non appartenente alla tabella interrogata e non voglio usare la clausola join.

1. Qual è la differenza tra l’operatore logico OR e l’operatore logico IN?

L’operatore OR pone due condizioni tra cui filtrare. Invece, l’operatore IN è una serie di espressioni logiche in cui ricercare la corrispondenza.

1. L’operatore logico BETWEEN include anche gli estremi del range specificato?

Sì, include anche gli estremi.

1. Che cos’è una SELF JOIN? Proponi un esempio.

La self join è quando una tabella viene collegata con sé stessa.

Ad esempio ho l’esigenza di sapere il nome del coniuge di una determinata persona.

Avrò la tabella Cittadino con i seguenti campi (CF, Nome, Cognome, DataNascita, LuogoNascita, CFConiuge).

Dovrò eseguire la seguente query:

*Select C.Nome, C.Cognome, Concat(CT.Nome, CT.Cognome) as Coniuge*

*From Cittadino as C*

*Self join Cittadino as CT*

*On C.CF =CT.CFConiuge*

**Case Study**

ToysGroup è un’azienda che distribuisce articoli (giocatoli) in diverse aree geografiche del mondo.

I prodotti sono classificati in categorie e i mercati di riferimento dell’azienda sono classificati in regioni di vendita.

In particolare:

1. Le entità individuabili in questo scenario sono le seguenti:

* Product
* Region
* Sales

1. Le relazioni tra le entità possono essere descritte nel modo seguente:

* Product e Sales
* Un prodotto puo’ essere venduto tante volte (o nessuna) per cui è contenuto in una o più transazioni di vendita.
* Ciascuna transazione di vendita è riferita ad uno solo prodotto
* Region e Sales
* Possono esserci molte o nessuna transazione per ciascuna regione
* Ciascuna transazione di vendita è riferita ad una sola regione

1. Le entità Product e Region presentano delle gerarchie:

* L’entità prodotto contiene, oltre alle informazioni del singolo prodotto, anche la descrizione della categoria di appartenenza. L’entità prodotto contiene quindi una gerarchia: un prodotto può appartenere ad una sola categoria mentre la stessa categoria può essere associata a molti prodotti diversi.

*Esempio: gli articoli ‘Bikes-100’ e ‘Bikes-200’ appartengono alla categoria Bikes; gli articoli ‘Bike Glove M’ e ‘Bike Gloves L’ sono classificati come Clothing.*

* L’entità regione contiene una gerarchia: più stati sono classificati in una stessa regione di vendita e una stessa regione di vendita include molti stati.

*Esempio: gli stati ‘France’ e ‘Germany’ sono classificati nella region WestEurope; gli stati ‘Italy’ e ‘Greece’ sono classificati nel mercato SouthEurope*.

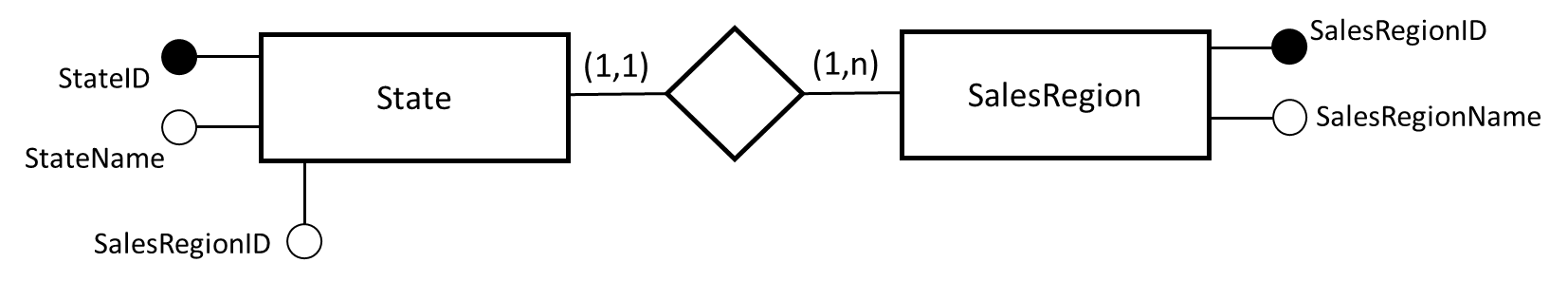
È necessario progettare e implementare fisicamente un database che modelli lo scenario garantendo l’**integrità referenziale** e la **minimizzazione della ridondanza dei dati.**

In altre parole, progetta opportunamente un numero di tabelle e di relazioni tra queste sufficiente a garantire la **consistenza del dato.**

**Task 1: Proponi una progettazione concettuale e logica della base dati**

La progettazione concettuale deve includere tutte le entità coinvolte e le relazioni tra queste. Per ciascuna entità indica l’attributo chiave e i principali attributi descrittivi (non è necessario indicare tutti gli attributi).

*Esempio di schema E/R*



*Lo schema proposto è puramente esemplificativo e non esaustivo o completo per la soluzione!*

La progettazione concettuale deve includere, per ciascuna tabella, tutte le colonne che poi verranno implementate fisicamente e deve esplicitare la cardinalità dei campi utilizzati per definire la relazione.

*Esempio di schema grafico delle tabelle e delle relazioni tra le stesse.*



n

1



*Il diagramma è puramente esemplificativo e non esaustivo o completo per la soluzione!*

***Descrizione attributi***

Product: IDProd, EnglishName, IDCat, Size, Price

Category: IDCat, CategoryName

Sales: SalesOrderNumber, OrderDate, IDState, IDProd, OrderQuantity, UnitPrice, SalesAmount

Region: IDRegion, RegionName

State: IDState, StateName, IDRegion

* *Progettazione concettuale*

**Immagine che contiene diagramma, schermata, viola, linea

Descrizione generata automaticamente**

* *Progettazione Logica*

|  |  |
| --- | --- |
| **Category** | |
| *IDCat* | *CategoryName* |
| 1 | Bike |
| 2 | Clothes |
| 3 | Swimming |
| 4 | Skiing |

**1**

**n**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | | | | |
| *IDProd* | *EnglishName* | *IDCat* | *Size* | *Price* |
| A1 | Kids' ST100 Mountain Bike | 1 | L | 150 |
| B2 | Synthetic Mountain Trekking Hooded Padded Jacket | 2 | L | 70 |
| C3 | Rockrider ST540 Mountain Bike 27.5" | 1 | M | 300 |
| D4 | Swimming suit | 3 | M | 30 |
| E5 | Freeride skis | 4 | L | 420 |
| F6 | Short Sleeve Cycling Jersey | 2 | S | 50 |
| G7 | Fins | 3 | S | 20 |
| H8 | FR900 MIPS Freeride Ski Helmet | 4 | M | 68 |
| I9 | Surface Mask | 3 | S | 30 |

**n**

**1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sale** | | | | | | |
| *SalesOrderNumber* | *OrdarDate* | *IDState* | *IDProd* | *OrderQuantity* | *UnitPrice* | *SalesAmount* |
| SO123 | 01-01-2022 | I | C3 | 1 | 300 | 300 |
| S0345 | 10-10-2023 | F | H8 | 3 | 68 | 204 |
| SO156 | 04-05-2023 | G | I9 | 4 | 30 | 120 |
| SO160 | 03-02-2021 | S | E5 | 1 | 420 | 420 |
| SO145 | 04-04-2023 | F | F6 | 2 | 50 | 100 |

**n**

**1**

|  |  |  |
| --- | --- | --- |
| **State** | | |
| *IDState* | *State* | *IDRegion* |
| I | Italy | SE |
| F | Francia | WE |
| G | Germany | WE |
| S | Spain | SE |

**n**

**1**

|  |  |
| --- | --- |
| **Region** | |
| *IDRegion* | *Region* |
| SE | South Europe |
| WE | WestEurope |

**Task 2: Descrivi la struttura delle tabelle che reputi utili e sufficienti a modellare lo scenario proposto tramite la sintassi DDL. Implementa fisicamente le tabelle utilizzando il DBMS MySQL (o altro).**

CREATE TABLE table\_name (ID

column1 datatype option,

column2 datatype option,

column3 datatype,

....

);

**Create table** Category (

IDCat int,

CategoryName varchar(60),

**CONSTRAINT** PK\_Product **PRIMARY KEY** (IDCat) );

**Create table** Product (

IDProd Varchar(2),

EnglishName varchar(50),

IDCat int,

Size Varchar(1),

Price int,

**CONSTRAINT** PK\_Product **PRIMARY KEY** (IDProd),

**CONSTRAINT** FK\_IDCat\_Product **FOREIGN KEY** (IDCat)

REFERENCES Category (IDCat)

);

**Create table** Sale

(SalesOrderNumber varchar(10),

OrderDate date,

IDState varchar(4),

IDProd varchar(2),

OrderQuantity int,

UnitPrice int,

SalesAmount int,

**CONSTRAINT** PK\_Sale **PRIMARY KEY** (SalesOrderNumber),

**CONSTRAINT** FK\_IDProd\_Sale **FOREIGN KEY** (IDProd)

**REFERENCES** Product (IDProd),

**CONSTRAINT** FK\_IDState\_Sale **FOREIGN KEY** (IDState)

**REFERENCES** State (IDState)

);

**Create table** State (

IDState varchar(3),

StateName varchar (20),

IDRegion varchar (3),

**CONSTRAINT** PK\_State **PRIMARY KEY** (IDState),

**CONSTRAINT** FK\_IDRegion\_Region **FOREIGN KEY** (IDRegion)

**REFERENCES** Region (IDRegion)

);

**Create table** Region (

IDRegion varchar (3),

RegionName varchar (20),

**CONSTRAINT** PK\_Region **PRIMARY KEY** (IDRegion)

);

**Task 3: Popola le tabelle utilizzando dati a tua discrezione (sono sufficienti pochi record per tabella; riporta le query utilizzate)**

insert into Category

values

(1, 'Bike'),

(2,'Clothes'),

(3,'Swimming'),

(4,'Skiing');

Insert into Product

Values

('A1', 'Kids ST100 Mountain Bike', 1, 'L', 150),

('B2', 'Synthetic Mountain Trekking Hooded Padded Jacket', 2, 'L', 70),

('C3', 'Rockrider ST540 Mountain Bike 27.5', 1,'M', 300),

('D4', 'Swimming suit', 3, 'M', 30),

('E5', 'Freeride skis', 4, 'L', 420),

('F6','Short Sleeve Cycling Jersey', 2, 'S', 50),

('G7', 'Fins', 3,'S', 20),

('H8', 'FR900 MIPS Freeride Ski Helmet', 4, 'M', 68),

('I9', 'Surface Mask', 3, 'S', 30);

insert into Sale

values

('SO123', '2022-01-01', 'I', 'C3', 1, 300, 300),

('S0345', '2023-10-10','F', 'H8', 3, 68, 204),

('SO156', '2023-05-04', 'G', 'I9', 4, 30, 120),

('SO160', '2021-02-03', 'S', 'E5', 1, 420, 420),

('SO145', '2023-04-04', 'F', 'F6', 2, 50, 100);

Insert into State

values

('I', 'Italy', 'SE'),

('F', 'Francia', 'WE'),

('G', 'Germany', 'WE'),

('S', 'Spain','SE');

Insert into Region

values

('SE', 'SouthEur'),

('WE','WestEur');

**Task 4: Dopo aver popolate le tabelle, scrivi delle query utili a:**

1. Verificare che i campi definiti come PK siano univoci. In altre parole, scrivi una query per determinare l’univocità dei valori di ciascuna PK (una query per tabella implementata).

Select IDProd

From Product

Group by IDProd

Having count(\*) >1;

Select IDCat

From Category

Group by IDCat

Having count(\*) >1;

Select IDState

From State

Group by IDState

having count(\*) > 1

Select IDRegion

From Region

Group by IDRegion

having count(\*) > 1

Select SalesOrderNumber

From Sale

group by SalesOrderNumber

having count(\*) > 1

1. Esporre l’elenco delle transazioni indicando nel result set il codice documento, la data, il nome del prodotto, la categoria del prodotto, il nome dello stato, il nome della regione di vendita e un campo booleano valorizzato in base alla condizione che siano passati più di 180 giorni dalla data vendita o meno (>180 -> True, <= 180 -> False)

Select SalesOrderNumber, OrderDate, EnglishName, CategoryName, StateName, RegionName,

Case when DATEDIFF(CURDATE(), OrderDate) > 180 then 'True' else 'False' end AS 'giorni>180'

from Category as C

inner join Product as P

On C.IDCat = P.IDCat

inner join Sale as S

on P.IDProd = S.IDProd

inner join State as ST

on S.IDState = ST.IDState

inner join Region as R

on ST.IDRegion = R.IDRegion

1. Esporre l’elenco dei prodotti che hanno venduto, in totale, una quantità maggiore della media delle vendite realizzate nell’ultimo anno censito. (ogni valore della condizione deve risultare da una query e non deve essere inserito a mano). Nel result set devono comparire solo il codice prodotto e il totale venduto.

Select S.IDProd, Sum(OrderQuantity) as QuantitàVenduta

from Sale as S

inner join Product as P

on S.IDProd = P.IDProd

Group by S.IDProd

having Sum(OrderQuantity) > (Select avg(orderquantity) as Media

from sale

where year(orderdate) = (Select max(year(orderdate))

from sale)

);

1. Esporre l’elenco dei soli prodotti venduti e per ognuno di questi il fatturato totale per anno.

Select EnglishName, Sum(SaleSAmount) as fatturato, Year(OrderDate) as anno

from Sale as S

inner join Product as P

on S.IDProd = P.IDProd

group by S.IDProd, year(OrderDate)

1. Esporre il fatturato totale per stato per anno. Ordina il risultato per data e per fatturato decrescente.

Select ST.StateName, Sum(SalesAmount) as fatturato, Year(OrderDate)

From State as ST

Inner join Sale as SA

on ST.IDState = SA.IDState

Group By SA.IDState, Year(OrderDate)

Order by Year(OrderDate), Sum(SalesAmount) desc

1. Rispondere alla seguente domanda: qual è la categoria di articoli maggiormente richiesta dal mercato?

Select CategoryName, Count(SalesOrderNumber) as Conteggio

from Sale as S

Inner join product as P

on S.IDProd = P.IDProd

inner join Category as C

on P.IDCat = C.IDCat

Group by CategoryName

Order by Count(SalesOrderNumber) desc

limit 1

1. Rispondere alla seguente domanda: quali sono i prodotti invenduti? Proponi due approcci risolutivi differenti.

Select IDProd, EnglishName

from product as P

where IDProd not in (Select IDProd

From Sale

Group by IDProd);

Approccio Join:

Select P.IDProd, EnglishName

From product as P

left join Sale as S

On P.IDProd = S.IDProd

where S.IDProd is null

1. Creare una vista sui prodotti in modo tale da esporre una “versione denormalizzata” delle informazioni utili (codice prodotto, nome prodotto, nome categoria)

Create view Prodotti as (

Select IDProd, EnglishName, CategoryName

from product as P

inner join Category as C

on P.IDCat = C.IDCat

);

1. Creare una vista per le informazioni geografiche

Create view Regione as (

Select IDState, StateName, RegionName

from State as S

join Region as R

on S.IDRegion = R.IDRegion

);